CS5500 : Reinforcement Learning

Assignment № 1

Due Date : 12/02/2020

Teaching Assistants : Shantam and Megha Gupta

Easwar Subramanian, IIT Hyderabad 01/02/2020

Assignment Policy

Read all the instructions below carefully before you start working on the assignment

- **Submission Guidelines**
  - Please include your name and institute roll number on the **first** page of the submission.
  - All answers should include **suitable justification**
  - We encourage to typeset the submissions using LaTeX or Jupyter notebooks with LaTeX embeddings. Scanned handwritten submissions, if submitted, must be clear and readable.
  - If any code is required as part of submission, submit them as Jupyter notebooks. In doing so, please make sure to include instructions to run the code and mention the python compiler version (2.7x or 3.x). The code should not contain any external dependencies unless otherwise specified as part of the question.
  - **Please submit your assignment answers as a private post to instructors/TAs under the label Assignment1** in Piazza by typing 'instructors' in the post to field
- **Collaboration Policy :**
  - It is OK to consult other students, course staff or browse internet to prepare answers for the problem. However, the submission has to be solely prepared by every individual without any plagiarism and with the **understanding** of the contents.
  - The graders reserve the right to conduct a mini viva to test the understanding of the answers in the submission. The result of the viva **will reflect in the final marks** of the overall assignment
  - The assignments are supposed to **help you understand the concepts better**. Hence, if you need help, please reach out to us during office hours or using Piazza.
- **Policy on Late Submissions :**
  - Assignments are normally expected on the due date.
  - However, we will allow a total of 6 late days across all four assignments.

Problem 1 : Markov Chains

Consider the following experiment in evolution. We are interested in the evolution of a gene by name **mTOR**. The gene occurs in two variants T and S and plays an important role in determining the height of an individual. Each individual has a pair of this gene, either TT (tall) or TS (medium height) or SS (short). In the case where the individual is of medium height, the order of gene pairing is irrelevant (TS or ST is same). In evolution, an offspring inherits a pair of gene from each of his/her biological parents with equal probability. Thus if one partner is tall (TT) and other partner is medium (TS), the offspring has $1/2$ probability of being tall or $1/2$ probability of being medium.

We shall start with an individual of a arbitrary but fixed trait (TT or TS or SS). The other partner is of medium height (ST or TS). The offspring of such partners again finds a medium height partner. This pattern repeats for a number of generations, wherein the resultant offspring always has a medium height partner.

(a) Write out the states and transition probabilities of this Markov process. (1 Point)

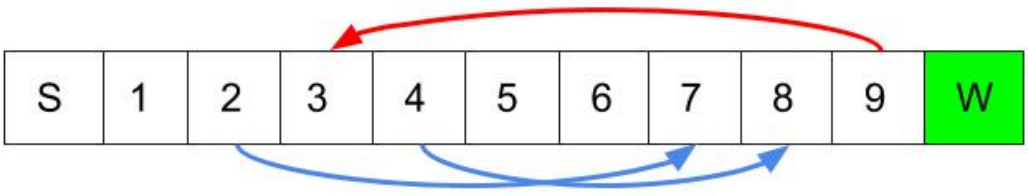
(b) Suppose we start with a medium height individual (as the first partner). What are the probabilities that any offspring belonging to first, second or third generation would be tall, medium or short in height ? (2 Points)

(c) What would be the answer to the previous question for any n-generations into future ? (1 Point)

[**Trivia and Disclaimer** : Actually, the height of an individual is not a single gene trait. But features like 'attached earlobes', 'short big toe' or 'PTC tasting' are examples of single gene traits. The purpose of this question is to test the understanding on Markov chains and not to drive home any biological concepts !!]

Problem 2 : Markov Reward Process

Consider the following snake and ladders game as depicted in the figure below.



- Initial state is S
- A fair four sided die is used to decide the next state at each time
- Player must land exactly on state W to win
- Die throws that take you further than state W leave the state unchanged

(a) Identify the states, transition matrix of this Markov process (2 points)

(b) From the transition matrix, pick out absorbing state(s), if any (1 point)

(c) Construct a suitable reward function, discount factor and use the Bellman equation for the Markov reward process to compute how long does it take "on average" (the expected number of die throws) to reach the state W from any other state (7 points)

[**Note** : Terminal reward at state W may have to be different from reward at other states]

Problem 3 : Markov Decision Process

A Lunar Roving Vehicle (LRV) is a battery operated four wheeled rover used on Moon to assist astronauts in their study of the lunar surface. The LRV is typically powered by solar energy. To tap solar energy into its panels, the LRV has to get on top of a hill. But the rover has a tendency to roll downhill and then it needs energy to ride uphill. Specifically, the LRV could be in three situations, namely, at the bottom of the hill, top of the hill or rolling down the hill. In each situation, the LRV could either drive or not drive. If the LRV is at the top of the hill and it is driving, at the next time instant, it could remain at the top of the hill with 0.8 probability or start rolling down the hill with 0.2 probability. If at the top of the hill and the LRV doesn't drive, these probabilities are 0.6 and 0.4 respectively. If the LRV is rolling down and drives, it will roll down with probability 0.4, end at hill top with probability 0.3 or end at the bottom with 0.3 probability in the next time period. If rolling down and doesn't drive, with probability 0.9 it will end at bottom and with 0.1 it will roll down in the next time period. Finally, if the LRV is at bottom hill and drives, it will roll down with probability 0.3 and end up hill with probability 0.7 in the next time step. If at the bottom of the hill and doesn't drive, the LRV will be at bottom of the hill with probability 1.

Further, driving always consumes 1 unit of energy. Also, the rover absorbs one, two and three units of energy while at the bottom, rolling down the hill and at top of the hill respectively. For example, if the LRV is at the top of a hill and driving the total reward for the LRV is, 3-1 = 2.

(a) Provide a graphical representation of the MDP [Refer to slide 19 of lecture 4]. (1 Point)

(b) Suggest a deterministic and stochastic policy for the MDP (It need not be optimal) (1 Point)

(c) For each policy suggested, write down the induced transition matrix and a corresponding Markov chain trajectory (1 Point)

(d) Suggest a history dependent policy to the MDP. (2 Points)

Problem 4 : Policy Evaluation and Partial Ordering of Policies

Consider the MDP shown in Figure 1. The MDP has 4 states $\mathcal{S} = \{A, B, C, D\}$ and there are two actions a_1 and a_2 possible. The actions determine which direction to move from a given state. We consider a stochastic environment such that action suggested by the policy succeeds

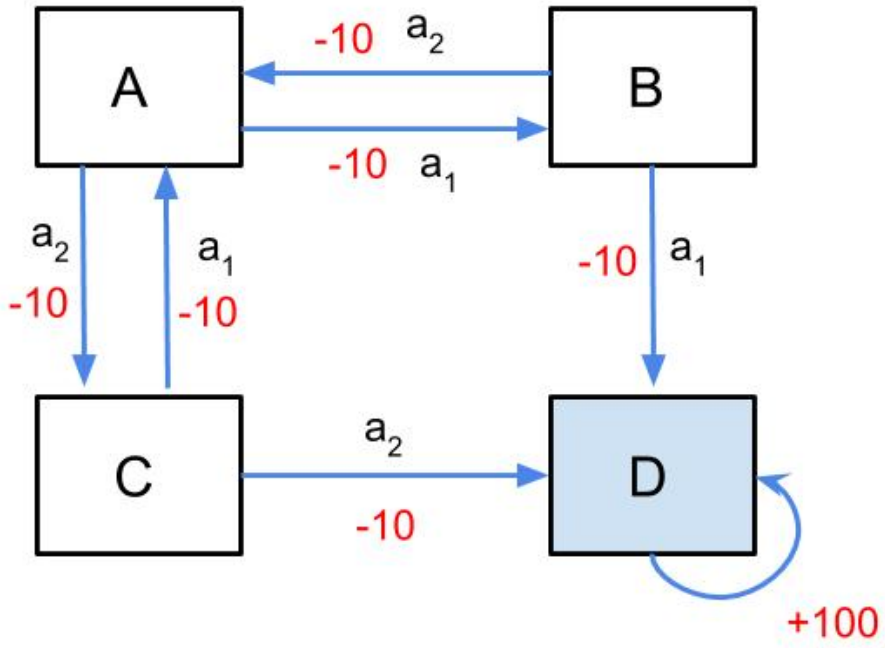


Figure 1: Partial Ordering of Policies

90 % of the times and fails 10 % of the times. Upon failure, the agent moves in the direction suggested by the other action. The state D is a terminal state with reward of 100. One can think that terminal states have only one action (an exit option) which gives the terminal reward 100. We consider three policies to this MDP.

- Policy π_1 is deterministic policy that chooses action a_1 at all states $s \in \mathcal{S}$.
- Policy π_2 is another deterministic policy that chooses action a_2 at all states $s \in \mathcal{S}$.
- Policy π_3 is a stochastic policy that chooses as follows
  - Action a_1 is chosen in states B and D with probability 1.0
  - Action a_2 is chosen in state C with probability 1.0
  - Action a_1 is chosen in state A with probability 0.4 and action a_2 is chosen with probability 0.6

(a) Evaluate $V^{\pi}(s)$ for each policy described above using the Bellman evaluation equation for all states $s \in \mathcal{S}$ (2 Points)

(b) Which is the best policy among the suggested policies ? Why ? (1 Point)

(c) Are all policies comparable ? Provide reason for your answer. (2 Points)

Problem 5 : Optimal Policies

A policy π_* of an MDP is said to be optimal if $\pi_* \in \Pi$ where

$$\Pi = \arg\max_{\pi} V^{\pi}(s) = \arg\max_{\pi} \mathbb{E}_{\pi}\left(r_{t+1} + \gamma r_{t+1} + \cdots | s_t = s\right)$$

Here the maximum condition has to be satisfied for all $s \in \mathcal{S}$ where $\mathcal{S}$ is the set of states of an MDP.

(a) Let us consider a finite state MDP with $|\mathcal{S}| = d$. We assume that for every state $s \in \mathcal{S}$ there exits at least one policy π such that $V^{\pi}(s)$ attains maximum for that given state s among all possible policies. That is, there could be different policies that attain maximum value for different states of the MDP. In such case, provide a way to construct an optimal policy π_* such that $V^{\pi_*}(s)$ would be maximum for all states of the MDP. (7 Points)

Problem 6 : Value Iteration

Consider the one dimensional grid world MDP shown in Figure 2. The MDP has six states

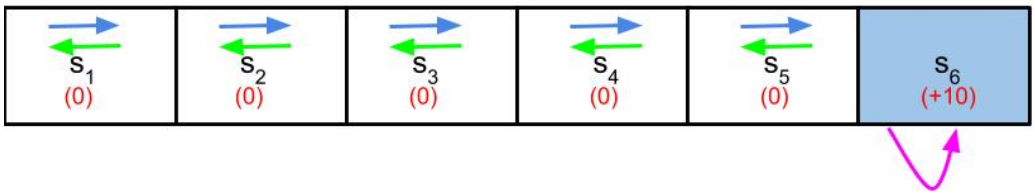


Figure 2: One Dimensional Grid World

$\mathcal{S} = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ and two actions $\mathcal{A} = \{L, R\}$ corresponding to moving left or right. The state s_6 is the terminal state with a reward +10. One can think of the terminal state as the state where the agent has only one action, that of the exit action, fetching a reward of +10. The environment is deterministic in the sense that all actions result in successful state transitions. Except that, the actions that take the agent of the grid, leaves the state unchanged. For answering parts of this problem and the next, you are free to implement value or policy iteration methods. But do provide intuitive reasoning to your answers along with the code, if any.

(a) Assume a discount factor of $\gamma = 1$. Find the optimal policy and optimal value function at every state including the goal state s_6 (1 Point)

(b) Now let the discount factor assume the following values $\gamma = \{0.9, 0.5, 0.1\}$. Study the impact on the optimal policy and optimal value function for each value of γ. Specifically, explain if the optimal policy remains the same for various values of the discount factor. (1 Point)

(c) We now consider adding a constant c to all rewards. The constant c can be positive or negative. Find the new optimal policy and optimal value function for all states including the goal state. Explain if the optimal policy would be different for different values of c (3 Points)

(d) We will now generalize the result that we got in the previous step. Let π be any policy of an MDP and V^{π} be the corresponding value function. Let $\widehat{V}^{\pi}$ be the value function of the policy π when a constant c is added to all rewards of the MDP. Derive an expression that relates V^{π} with $\widehat{V}^{\pi}$ (5 Points)

Problem 7 : Effect of Noise and Discounting

Consider the grid world problem shown in Figure 3. The grid has two terminal states with positive

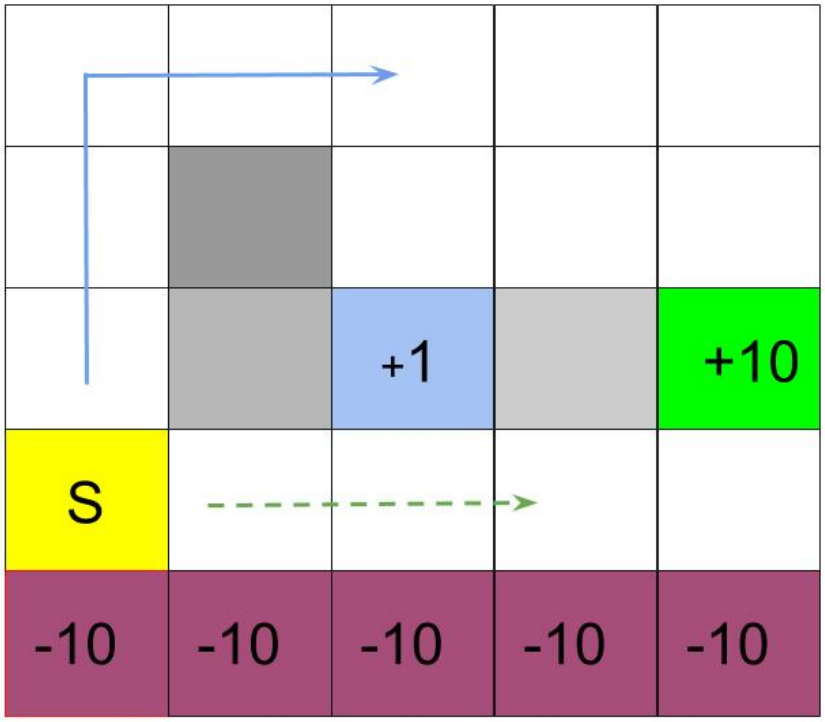


Figure 3: Modified Grid World

payoff (+1 and +10). The bottom row is a cliff where each state is a terminal state with negative payoff (-10). The greyed squares in the grid are walls. The agent starts from the yellow state S. As usual, the agent has four actions $\mathcal{A} =$ (Left, Right, Up, Down) to choose from any non-terminal state and the actions that take the agent off the grid leaves the state unchanged. Notice that, if agent follows the dashed path, it needs to be careful not to step into any terminal state at the bottom row that has negative payoff. There are four possible (optimal) paths that an agent can take.

- Prefer the close exit (state with reward +1) but risk the cliff (dashed path to +1)
- Prefer the distant exit (state with reward +10) but risk the cliff (dashed path to +10)
- Prefer the close exit (state with reward +1) by avoiding the cliff (solid path to +1)
- Prefer the distant exit (state with reward +10) by avoiding the cliff (solid path to +10)

There are two free parameters to this problem. One is the discount factor γ and the other is the noise factor (η) in the environment. Noise makes the environment stochastic. For example, a noise of 0.2 would mean the action of the agent is successful only 80 % of the times. The rest 20 % of the time, the agent may end up in an unintended state after having chosen an action.

(a) Identify what values of γ and η lead to each of the optimal paths listed above with reasoning. If necessary, you could implement the value iteration algorithm on this environment and observe the optimal paths for various choices of γ and η. (10 Points)

[**Hint** : For the discount factor, try high and low γ values like 0.9 and 0.1 respectively. For noise, consider deterministic and stochastic environment with noise level η being 0 or 0.5 respectively]

Problem 8 : Convergence Rate of Value Iteration

In this problem, we will derive an expression to understand the convergence rate of value iteration algorithm. Specifically, we will consider the value iteration that finds the optimal value function V_*. Recall the following notations. The Bellman update equation used in the value iteration algorithm is given by

$$V_{k+1}(s) \leftarrow \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_k(s') \right) \right].$$

In the case we assume the environment to be deterministic, the above update rule simplifies to,

$$V_{k+1}(s) \leftarrow \max_a \left[\mathcal{R}^a_{ss'} + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} V_k(s') \right].$$

We define the Bellman optimality operator to be

$$L(v) = \max_{a \in \mathcal{A}} [\mathcal{R}^a + \gamma \mathcal{P}^a v] \quad (\text{ Slide 19 of Lecture 8})$$

It is easy to see that V_* is a fixed point of operator L as

$$V_*(s) = \max_a \left[\mathcal{R}^a_{ss'} + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} V_*(s') \right] \quad (\text{ Slide 20 of Lecture 5})$$

We now start the value iteration algorithm with an initial guess V_1 and let V_{k+1} be the $k+1$-th iterate of V in the value iteration algorithm. Derive an expression that computes the distance between $k+1$-th iterate of V and the optimal value function V_*. (using the max norm). Specifically, provide an analytical expression for $|V_{k+1} - V_*|_\infty$. Use the expression obtained to prove that the value iteration converges geometrically, i.e.

$$|V_{k+1} - V_*|_\infty \leq \gamma^k |V_1 - V_*|_\infty$$

where $\gamma < 1$ is the discount factor of the MDP. (8 Points)

Problem 9 : Programming Value and Policy Iteration

Implement value and policy iteration algorithm and test it on **'Frozen Lake'** environment in openAI gym. **'Frozen Lake'** is a grid-world like environment available in gym. The purpose of this exercise is to

- Get hands on with using gym.
- Help understand the implementation details of value and policy iteration algorithm(s)
- You may use the code developed for this question to answer relevant parts of problem 6 and 7

This question will not be graded but **will** still come in handy for future assignments.

ALL THE BEST